APPENDIX - D

BINDING AND BIND SEQUENCE NUMBER :

Consider an Rcp Gate G1, with the following configuration :

G1{N1[2], L1} → {QA1[8]}

Since there are no inputs, the Rcp Gate G1 will select all available output queues in the output queue array QA1, and binds them to itself, which comprises of storing the bindings in the bind table of the Rcp gate. When an output queue is bound, the Rcp gate obtains a unique sequential number form the local ring to which it is connected, and assigns the unique sequential number to the binding by storing it in the bind table entry. This unique sequential number is called the BIND SEQUENCE NUMBER.

It may be noted that the bind sequence number is required to track the inputs which arrive on the input side of an Rcp Gate. Since there can be multiple invocations running concurrently for the top level Rcp Gates, the lower level Rcp Gates can see their inputs out of order, however the BIND SEQUENCE NUMBER helps the receiving Rcp Gate to reorder the inputs, so that the original order is not lost.

Another major purpose of the bind seq number is to provide a layer over the queues, so that any available queue set may be used, for a particular bind seq num. For example, bind seq num 33, need not use the 1st queue in the output queue arrays(assuming that the bind table size is 32). Bind seq num 33, may use any output queue set that is available at that time, which may be 10.

It may be noted that BIND SEQUENCE NUMBER may not be misinterpreted as an equivalent for the token concept of the data flow computing. Since the tokens of data flow computing architecture and the BIND SEQUENCE NUMBER of Rcp architecture are quite different and serve very different purposes. It is very important to observe that BIND SEQUENCE NUMBERS are not used for matching, and that there is no concept of matching in the Rcp architecture. In other words, data flow tokens are used for matching, whereas Rcp BIND SEQUENCE NUMBERS are used for sorting. The stage two of the BIND_VIRTUAL_QUEUES function 3207, in fact rearranges the inputs it receives, and this can be considered as a special kind of sorting.

With regard to the lack of matching in Rcp architecture, it may be noted that the Rcp architecture cannot handle certain configurations, which are described below.

Assume Rcp Gate G1, and Rcp Gate G2, and Rcp gate G3 are defined with the following configurations.

G1{N1[1], L1} → {QA1[8]}
G2{N2[1], L2} → {QA2[8]}
{QA1, QA2} → G3{N3[1], L3} → {QA3[8]}

The configuration for Rcp gate G3 is invalid because the Rcp gates G1 and G2 are independent of each other, and their outputs are not synchronized.

Associative matching is very well known in prior art, and hence is not discussed in this document. Implementers may optionally fuse other well known concepts with Rcp architecture.

The life cycle of the BIND SEQUENCE NUMBER is provided below for convenience.

1) Local Ring maintains the BIND SEQUNCE NUMBER, so that multiple Rcp gates writing output to the same queue array, still get sequential numbers, since Rcp gates which have common output queue arrays are connected to the same local ring.
2) Rcp Gate locks the local ring and acquires the BIND SEQUENCE NUMBER, when it allocates an output queue set identified by the output queue index to the incoming non null input queue set, identified by the input queue index. The incoming input queue set will have a BIND SEQUENCE NUMBER (qualified as INPUT), which may be different from what is allocated by this Rcp Gate (qualified as OUTPUT), since null inputs are bypassed by the Rcp Gate.
3) The Rcp gate stores the binding info, the input/output queue indexes, and the BIND SEQUENCE NUMBER in the bind table entry.
4) When a node function invocation does a successful Rebind, the BIND SEQUENCE NUMBER (OUTPUT) is copied to the node function invocation structure.
5) When this node function invocation sets an output queue to ready state, the BIND SEQUENCE NMBER is copied from the node function invocation structure to the Bind Sequence table entry of the Queue Array, corresponding to the queue number.
6) When the queue is consumed by all consumers, the RESET_QUEUE function 3210, sets the BIND SEQUENCE NUMBER in the Bind Sequence table entry of the Queue Array to NULL.
7) The BIND SEQUENCE NUMBER in the node function invocation structure is copied over by a new value during next iteration.
8) The BIND SEQUENCE NUMBER stored in the Bind table of the Rcp gate is reset to NULL by the Unbind_Virtual_Queues 3209 function.